REMARKS

Claims 1-10, 13-21 and 25 were examined in the Final office action mailed on 05/28/2008 (hereafter "Second Final Office Action"). The specification was objected to and all the claims were rejected. Applicants respectfully request reconsideration further in view of the below remarks.

Specification

The abstract of the disclosure was objected to requiring that the paragraph number there be removed. The Abstract is accordingly sought to be amended. Withdrawal of the objection as against the specification is respectfully requested.

Claim Rejections Under 35 U.S.C. § 103

Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Gostanian *et al*, U.S. 5,781,910 (Gostanian), in view of "XA Components, Oracle8*i* JDBC Developer's Guide and Reference, 1999" (JDBC Guide).

Without acquiescing to the Examiner's contentions, it is submitted that the previously presented claims are allowable over the art of record at least for the reasons noted below..

For example, as noted in the previous response, previously presented independent claim 1 recites, among other features, that:

1. "... requesting in a user program a transaction identifier for an atomic transaction"; and

2. "... specifying in said user program a plurality of combinations, wherein each of said plurality of combinations contains said transaction identifier, a task procedure, and a rollback procedure, wherein said task procedure implements a part of said atomic transaction and said rollback procedure is designed to rollback said task procedure; ... wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program, ..."

For the benefit of the Examiner, it is pointed out that the claimed user programs are illustrated with reference to Figure 2 of the subject application. The user program there is

shown requesting a transaction identifier in line 209. The claimed combinations including the transaction identifier, the task procedure (forming a part of the atomic transaction) and the corresponding rollback procedure are shown in each of the lines 215, 225, 255, 260, 265 and 270. As noted above, each of these lines shows a task procedure and a rollback procedure as separate procedures.

The construction of the above features implies that the **users** designing the user programs **have the ability to specify custom roll back procedures and task procedures**.

Gostanian and JDBC Guide, neither individually nor in combination, teach or reasonably suggest the above noted features.

The claimed user programs are akin to application code 624 of Gostanian, as will be apparent to a skilled practitioner by reading the disclosure of the subject application, as well as the below disclosure:

> The application clients 602, 604 preferably include at least three layers. First, the ***application client 602 includes a user-defined application code layer 624.*** The application code 624 may be generated by commercially available tools, such as Microsoft Access.RTM. or may be written in a standard computer language, such as C, to perform the user-defined functions. In order to manipulate data items stored at the database servers 612, 614, the application code 624 issues ODBC API calls to an ODBC driver manager layer 626. The driver manager layer 622, which is essentially a pass-through, directs all ODBC API calls issued by the application code 624 to an ODBC Client layer 628. The ODBC Client layer 628 is a universal driver and typically runs on 32-bit Microsoft Windows-based client machines (e.g., Windows NT.RTM. or Windows 95.RTM.). It is the ODBC Client layer 628 that actually sends the ODBC API calls across the network 610 to the database sites 612, 614. ***It should be understood that each application client 602, 604 may run on a client computer 220 (FIG. 2)***.
> (Col. 187, lines 3-21 of Gostanian, ***Emphasis Added***)

The application clients (which are user defined) of Gostanian do not have the specific claimed intelligence for implementing atomicity of transactions.

Rather, in Gostanian, **common** underlying components **alone** perform concurrent transactions in a replicated database environment, at least based on the below:

> Associated with each database site 312, 314 is an application server 332, 334, respectively. As illustrated in FIG.

> 3, the application servers 332, 334 may be layered over their respective database sites 312, 314. The application servers 332, 334 coordinate the requested database transactions for the application clients 302-308. Specifically, the application clients 302-308 request the execution of transactions preferably by issuing application programming interface (API) calls via the communications network 310 to the application servers 332, 334. The ***application servers 332, 324 then communicate with each other as described below and with their corresponding database servers 312, 314***, through conventional RDBMS API calls, such as ORACLE Pro*C.RTM. or SYBASE Embedded SQL/C.RTM. precompiler statements in order ***to update the relevant data items in response to the requested transactions***.
>
> More specifically, each application server 332, 334 preferably includes a manager process 336, 338 and one or more worker processes 340, 342. The manager processes 336, 338 communicate with each other over the communications network 310 and act as the control centers handling requests from and replies to the application clients 302-308. ***The manager processes 336, 338 also preferably supervise the replication of data across the system 300 and coordinate the commits and aborts in the corresponding database servers 312, 314***. The worker processes 340, 342 communicate directly with their associated database servers 312, 314, respectively. That is, the RDBMS API calls are issued by the worker processes 340, 342 to their associated database management systems 316, 318.
>
> (Col. 9 lines 27-56 of Gostanian, ***Emphasis Added***)

From the above, it is respectfully submitted that the **manager processes 336/338** coordinate the commits and aborts, and all the application clients rely on the manager processes for the commits/aborts (in general to implement the concurrent transactions in a replicated database environment).

It is accordingly concluded that the application clients of Gostanian are akin (at least implement) to the claimed user programs, and there is no disclosure or reasonable suggestion in Gostanian to incorporate the claimed intelligence into the application clients of Gostanian. The Examiner appears to agree as well by relying on JDBC Guide instead.

JDBC Guide does not cure the above noted deficiency.

In particular, JDBC Guide teaches

> XA Connection Interface and Oracle Implementation
>
> ***An XA connection instance***, as with a pooled connection instance, ***encapsulates a physical connection to a database***. This would be the database specified in the connection properties of the XA data source instance that

> produced the XA connection instance. (Page 2 lines 3-5 of JDBC Guide, ***Emphasis Added***)
>
> ...
>
> XA Resource interface and Oracle Implementation
>
> ***The transaction manager uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction.***
>
> Each XA resource instance provides the following key functionality, typically invoked by the transaction manager:
>
> o It associates and disassociates distributed transactions with the transaction branch operating in the XA connection instance that produced this XA resource instance, (Essentially, associates distributed transactions with the physical connection or session encapsulated by the XA connection instance.) This is done through use of transaction Ids. (Page 2 lines 3-5 of JDBC Guide, ***Emphasis Added***)

From the above, it is pointed out that JDBC Guide envisages a 'transaction manager' which uses XA resource instances to coordinate all the transaction branches that constitute a distributed transaction.

It is asserted that the transaction manager again is a **central entity**, which is used by several end applications, and there is no disclosure or suggestion that the transaction manager is present in each of the end applications interacting with the database.

Accordingly it is contented that Gostanian and JDBC Guide, neither individually nor in combination, teaches several features of previously presented claim 1.

Further, since each of Gostanian and Oracle teaches a alternative solution to a potentially similar objective as in claim 1, each reference individually teaches away from previously presented claim 1.

At least for such reasons, previously presented claim 1 is allowable over the art of record.

Currently amended claim 7 is also allowable for at least some of the reasons noted above in reciting that, "***requesting an identifier in said user program from a transaction manager*** for said atomic transaction, wherein said transaction manager generates a unique

value as said identifier; setting a variable to equal said identifier; specifying a plurality of combinations ***in said user program*** for execution in said system, wherein each of said plurality of combinations ***contains said variable transaction identifier, a task procedure, and a rollback procedure***,..." (***Emphasis Added***)

Again, since the application programs of both Gostanian and JDBC Guide rely on common components for all the applications implementing atomic transactions, the references relied upon by the Examiner would neither request an identifier nor specify (using a variable) an identifier along with task procedures **within a user program**.

Accordingly, currently amended claim 7 is also allowable over Raz.

Currently amended independent claims 10 and 16 are allowable over the art of record at least in reciting, "... wherein said rollback procedure is specified as a separate procedure from said task procedure in said user program."

Accordingly all the independent claims presented for consideration are allowable over the art of record. The dependent claims are allowable at least as depending from the corresponding base claims.

Conclusion

Thus, it is believed that all rejections have been overcome. The Examiner is respectfully requested to withdraw the final rejection and continue examination. The Examiner is invited to telephone the undersigned representative at 707.356.4172 if it is believed that an interview might be useful for any reason.

Respectfully submitted,

/Narendra Reddy Thappeta/
Signature
Printed Name: Narendra Reddy Thappeta
Attorney for Applicant
Registration Number: 41,416

Date: July 17, 2008